Parallel 4-Dimensional Cellular Automaton Track Finder for the CBM Experiment

Akishina V., Kisel I.

Goethe-Universität Frankfurt, Frankfurt am Main, Germany
Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany
GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany

DPG Spring Meeting, Darmstadt
17.03.2016
Reconstruction Challenge in CBM

Correct procedure of event building from time-slices is crucial for right physics interpretation.
First Level Event Selection Package (FLES)

Event building as a part of FLES package
Cellular Automaton (CA) Track Finder

0. Hits

1. Segments

2. Neighbors & Counters

3. Track Candidates

4. Tracks

Detector layers

Hits

Cellular Automaton:
• local w.r.t. data
• intrinsically parallel
• simple
• very fast

Perfect for many-core CPU/GPU!

Cellular Automaton:
1. Build short track segments - triplets.
2. Mark possible neighbours while building triplets.
3. Connect according to the track model, estimate a possible position on a track.
4. Tree structures appear, collect segments into track candidates.
4. Select the best track candidates.
Time-based CA Track Finder

How to use time information in tracking?

- Triplets are build from the hits with the same time measurement within $3 \sigma$ of detector precision
- Fast access to the hits is provided by time-based structure: hits are sorted by time and space coordinates and stored into the time-based grid

Variable time step of grid: 4D tracking in a 3D-style approach
Parallel implementation with OpenMP and Pthreads

<table>
<thead>
<tr>
<th>Algorithm Step</th>
<th>% of total execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialisation</td>
<td>8%</td>
</tr>
<tr>
<td>Triplets construction</td>
<td>64%</td>
</tr>
<tr>
<td>Tracks construction</td>
<td>15%</td>
</tr>
<tr>
<td>Final stage</td>
<td>13%</td>
</tr>
</tbody>
</table>

Total time = 84 ms

Total time = 849 ms

Speed-up factor 10.1 Theoretically achievable factor: 13
### 4D Track Finder in CBMroot Framework

<table>
<thead>
<tr>
<th>Efficiency, %</th>
<th>3D</th>
<th>4D</th>
<th>CBMROOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>All tracks</td>
<td>92.1</td>
<td>92.2</td>
<td>91.3</td>
</tr>
<tr>
<td>Primary high-p</td>
<td>97.9</td>
<td>97.9</td>
<td>99.1</td>
</tr>
<tr>
<td>Primary low-p</td>
<td>93.6</td>
<td>93.5</td>
<td>93.6</td>
</tr>
<tr>
<td>Secondary high-p</td>
<td>92.0</td>
<td>92.0</td>
<td>88.9</td>
</tr>
<tr>
<td>Secondary low-p</td>
<td>65.7</td>
<td>65.9</td>
<td>56.8</td>
</tr>
<tr>
<td>Clone level</td>
<td>2.8</td>
<td>3.1</td>
<td>3.7</td>
</tr>
<tr>
<td>Ghost level</td>
<td>4.9</td>
<td>4.2</td>
<td>1.9</td>
</tr>
<tr>
<td>Time/event/core</td>
<td>11.7 ms</td>
<td>13.6 ms</td>
<td>17.3 ms</td>
</tr>
</tbody>
</table>

CBMroot revision 8357 (Nov 2014)

3D, 4D: AuAu 25 AGeV mbias events at 10MHz
CBMROOT: AuAu 10 AGeV mbias events at 10MHz

Time-based tracking performance comparable with event-by-event
Reconstructed tracks are clearly clustered in groups representing original events.
Event Building at IR = 10 MHz

Reconstructed tracks are grouped in events using histogramming:
- all tracks are filled in a time histogram with bin width of 1 ns
- neighbouring not empty bins are called an event
- gap of a 4 empty bins is a sign for event end

- 70 reco events are reconstructed one-to-one, 7 reco events are merged together.
- Primary tracks can be separated using primary vertex information.
- Search of only one primary vertex per event using KF Particle Finder package is currently implemented.
- Multi-vertex reconstruction is in progress.

7% of events are merged to be studied with multi-vertex analysis
Summary

• Event building is a necessary part of FLES package
• Time-based 4D track finder allows to reconstruct time-slices with speed and efficiency comparable to event-based approach
• 4D track finder is parallel with speed-up 10.1 out of 13 theoretically achievable within the Intel Xeon E7-4860 CPU
• A first version of event building was implemented based on the 4D tracking.

Future Plans
• Multiple primary vertices analysis
• Physics analysis